PRIORIZATION OF MANAGEMENT OBJECTS

FIELD OF THE INVENTION

[0001] The invention relates to priorization of management objects in a device management system.

BACKGROUND OF THE INVENTION

[0002] As different data processing devices, such as mobile phones, are made more complicated, the significance of device management becomes more important. Various settings are required in such devices, for instance settings associated with Internet access points, which are difficult and time-consuming to set manually. Device management solutions have been created in order to solve such problems. The device management solutions allow, for example, an operator of a company's information system or a teleoperator to provide the device with an appropriate configuration. Generally, device management refers to measures that enable external sources of the device to change the configuration of the device, for instance to change the settings or even the protocol the device employs. In addition to the settings associated merely with the device, user-specific data can also be sent, such as user profiles, logos, ringing tones and menus, which allow the user to personally modify the settings of the device or the modification is carried out automatically in connection with device management.

[0003] A device management standard is SyncML (Synchronization Markup Language) device management, which is partly based on SyncML data synchronization standard enabling data synchronization. A synchronization server may operate as a device management server and a customer device may operate as a device management client. As to the device management, the customer device functioning as the customer provides the management server performing device management in the starting message of the session with data about itself (the same data as during synchronization) to the server, the management server responding to the data by sending specific data and server management operations. The customer device responds with status information, whereafter the server may end the session or send more server management operations. If the server provides more management operations, the customer device should respond with status information. After receiving status information, the server may end the session or continue the session by providing more server management operations. The device man-

agement protocol may also operate so as to provide the user at first with questions about what he/she wants to update, the server then being informed about the choices of the user. The server may thereafter send the updates/operations desired by the user in the following packet.

[0004] The objects to be managed in a customer device are arranged as management objects. The management objects constitute of entities in the customer device, which are manageable using the management operations of the management server. A management object may be, for instance, a parameter value or a large entity, such as a wallpaper or screen saver. The management objects are arranged in the SyncML device management into a form resembling a tree in a management tree illustrated in Figure 1. The management object may be an individual parameter, a sub-tree or a data collection. For example, a 'Vendor' management object is a node, or an interior object, since it is provided with child objects 'Screen Saver' and 'Ringing Tones'. The management object 'Screen Saver' is a leaf object, since it has no child objects. The management object 'Ringing Tones' is also a node or an interior object, because it is provided with child objects. The contents of a management object may also include a link, which indicates another management object. Each object can be indicated using a URI identifier (Uniform Resource Identifier). The URI of the management object is formed starting from a root '/' and as the process proceeds along the tree, each management object is provided with a name that is added to the previous ones using the letter '/' as a distinguishing mark. For example, the management object 'Ringing tones' can be identified using the URI identifier '/Vendor/Ringing Tones/'. Preferably, at least some of the management objects are standardised (the SyncML device management standards currently include three standardised management objects). The management objects may be fixed or dynamic. Dynamic management objects may be added to the management tree from a customer device or management server.

[0005] The new dynamic management object must be indicated with a name (address) in the management tree so that the management tree is the same both in the management server and in the customer device. An example of a new management node is a document provided with WAP Provisioning Settings. A new management object can be formed of the document and the data elements included therein, and the management object is provided with new nodes or different settings as child objects. The WAP provisioning docu-

ment is an XML-based (Extensible Markup Language) file, which comprises different parameter sets associated with the WAP provisioning settings that are consecutively listed. The WAP provisioning document may also comprise parameter sets, which may be mutually alternative, for example, several optional transfer servers or transmission path settings of the network level. In devices that do not employ a management system, the WAP provisioning document is stored as a single document, from which parameter values are read if need be.

[0006] What becomes a problem in devices that employ a management system is deassembling the WAP provisioning document and placing parameter sets included therein as well as attributes included therein as management objects into the above management tree. The attributes may be placed into the management tree in a completely different order than their mutual relations require. This is because the SyncML definition does not offer any means for determining the order of the parameter sets and the attributes included therein and for defining the mutual priority based on the WAP provisioning document. In the WAP provisioning document, the priority of the parameter sets and the attributes included therein is defined according to the order thereof. A need therefore arises for a method that determines for instance the priority order of mutually alternative parameter sets when such parameter sets are deassembled from the WAP provisioning document to the management tree.

SUMMARY OF THE INVENTION

[0007] Now it has been invented an improved method and an apparatus implementing the method for forming the management tree. As different aspects of the invention a method, a device management system, an electronic device and computer software products are introduced, which are characterized in what is disclosed in the independent claims.

[0008] The preferred embodiments of the invention are disclosed in the dependent claims.

[0009] In accordance with the invention, priority is determined for a management object and sub objects belonging thereto in a device management system comprising at least a device management server and a management customer device, in which the management server and the management customer device are arranged to update management object information in a management tree. The contents of the sub objects belonging to at least one

management object in the management tree are determined, and thereafter at least one data element is determined in the management server, the data element comprising the priority data of at least one sub object in relation to other sub objects. Thus at least one data element is attached to the management tree maintained by the management server, and the file definition according to the management tree is sent to the management customer device. The management customer device deassembles the received file definition into the management tree form so that the priority data shows at least the priority data of at least one sub object in relation to other sub objects.

[0010] The method of the invention is particularly applicable, but not limited, to a server device according to SyncML Device Management and a customer device according to SyncML device management, in which priority data is determined for the management object comprising Provisioning Settings of a WAP protocol for a Bootstrap process.

[0011] In accordance with a embodiment of the invention, the data element is determined in the management server, said data element comprising the priority data of at least one sub object in relation to other sub objects, as separate leaf objects, which are attached to the management tree maintained by the management server so that they are placed in parallel with the management/sub object, whose priority they determine.

[0012] In accordance with another embodiment, the data element is determined in the management server, said data element comprising the priority data of at least one sub object in relation to other sub objects, as a run- time property (RTProperty) definition, which is attached to the meta data of the management tree maintained by the management server.

[0013] The management object generally refers to an entity associated with the device management of a device to be configured, such as the settings of an Internet access point, without being restricted to the management objects of the SyncML device management. What a data element generally refers to is a management object record provided with a predetermined identifier, such as a data element according to XML language containing at least one or more sub-elements and/or values (in any mode determined for an element). Device management is not restricted to the operations defined in SyncML, but they must be broadly understood as device management to be carried out between any manageable customer device and device management server and as messages required therein.

[0014] The solution of the invention provides such an advantage that the method allows deassembling previously problematic management objects and placing the parameter sets and attributes included therein as management objects into the management tree so that the priorities among the management objects are taken into account. Another advantage is that the management server does not have to check afterwards from the management customer device, whether the management objects are placed in the management tree in the correct order. This is an advantage of particularly great importance in the operation of the Bootstrap process, where the management customer device does not provide the management server with an acknowledgement.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] In the following the invention will be described in greater detail by means of the embodiments with reference to the accompanying drawings, in which:

Figure 1 illustrates a management tree;

Figure 2 illustrates a management system;

Figure 3 illustrates a server and a customer device;

Figure 4 illustrates a method according to a first embodiment of the invention; and

Figure 5 illustrates a method according to a second embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0016] In the following, a preferred embodiment of the invention is described in a system supporting SyncML device management. However, it should be noted that the invention could be applied in any device management system, where management objects of a device can be organized in other ways than merely as a tree structure.

[0017] Figure 2 illustrates a networked system. A network server or PC typically operates as a server S. Typically, the terminal TE is a mobile phone, a PC (Personal Computer), a laptop computer, or a PDA device. For clarity, it is noted in the following embodiments that regarding device management, the terminal TE operates as a customer device and the server S as a management server. The server S may manage several customer devices TE. However, it is possible that the terminal TE carries out the functionalities of the

management device, whereby another terminal or server S may function as the customer device.

[0018] Figure 2 shows two examples. In the first example, customer devices TE and management servers S are connected to a local area network LAN. The customer device TE connected to the network LAN comprises a functionality, for instance software controlling a network card and data transmission, for communicating with the devices in the network LAN. The local area network LAN may be a local network of any kind and the TE may communicate with the server S also through the Internet, typically using a firewall FW. The terminal TE may also be connected to the local area network LAN wirelessly through access points AP. In the second example, the customer device TE communicates with the server S through a mobile network MNW. The terminal TE connected to the network MNW comprises a mobile message functionality for wirelessly communicating with the network MNW. In addition, other networks such as the local area network LAN can be placed between the mobile network MNW and the server S. The mobile network MNW may be any already known wireless network, for instance a network supporting the GSM service, a network supporting the GPRS service (General Packet Radio Service), a third generation mobile network such as a mobile network according to the 3rd Generation Partnership Project network definitions, a wireless local area network WLAN, a private network or a combination of such networks. An important service of the transport layer in several mobile networks is WAP, which offers by means of a Wireless Session Protocol WSP layer included therein a transport service for the device management application layer in the customer device TE and the server S. Then, the system comprises at least one WAP gateway and possibly one or more WAP Proxy servers. The WAP supports many transmission techniques of lower layers, such as the HTTP or OBEX standards. For example, circuit-switched or packet-switched data transmission or SMS-based transmission in accordance with the mobile network MNW properties can be employed as the lower layer transmission techniques. In addition to the examples shown in Figure 2, other device management configurations are also possible.

[0019] As Figure 3 illustrates, the terminal TE and the server S comprise memory MEM; SMEM, a user interface UI; SUI, I/O means I/O; SI/O for arranging data transmission, and one or more central processing units CPU; SCPU comprising a processor. The memory MEM; SMEM includes a

non-volatile portion for storing the applications controlling the central process-ing unit CPU; SCPU and other data to be stored and a volatile portion to be used for temporary data processing. The memory MEM of the terminal TE stores management objects, the structure of which is also maintained in a management tree in the memory SMEM of the server S.

[0020] The TE operating as a customer device according to the SyncML device management standard comprises a customer agent CA, which attends to the operations associated with the management session in the cus-tomer device. The device S operating as a management server comprises a server agent SA handling the management session. The customer agent CA may be implemented by performing in the CPU computer software code stored in the memory MEM and the SA by performing in the SCPU computer software code stored in the memory SMEM. As noted previously, the TE and the S may operate as a management server and/or as a customer device. Thus, for ex-ample the terminal TE may also comprise at least partly the functions of the server agent SA, in which case it may also operate as a management server during data transmission between the terminals TE. The computer software codes to be carried out in the central processing units CPU and SCPU may allow the terminal TE and the server S to also implement the inventive stages associated with addressing the priority of the management objects, some of the preferred embodiments thereof being illustrated in Figures 4 and 5. The computer software may be stored into any memory means, such as the hard disk of a PC or a CD-ROM disc, from where it can be loaded into the memory MEM; SMEM of the implementing device TE; S. The computer software can also be loaded through a network, for instance using a TCP/IP protocol stack. It is also possible to use hardware solutions or a combination of hardware and software solutions for implementing the inventive means.

[0021] The management objects are typically based on XML lan-guage. A document type definition DTD is defined for XML-based documents. The DTD defines in XML language the tags to be used, the structural relations of the elements between the tags, the attributes defining the properties of the elements and the parameters describing the attributes and the values thereof. As is known in the XML language, a data element starts from an initial tag (for instance <section>) and ends at the end tag (</section>), and the tag may contain text as well as other elements. The DTD is involved in all documents or the documents refer to an already known DTD.

[0022] As described above, the method of the invention is particularly applicable to transfer WAP provisioning settings to the terminal TE. The method is particularly useful in the Bootstrap process, in which the required application settings and general device-specific settings, such as one-directional WAP settings, are sent to the un-configured terminal TE. The Bootstrap process thus deviates from typical SyncML device management, in which a status is sent from the terminal. The application settings and general device-specific settings, such as WAP settings, can be transferred utilizing the WAP protocol to the terminal, and thereafter the new management objects according to the WAP provisioning settings must be added to the management tree of the SyncML device management protocol. In order to be able to deassemble the WAP provisioning document and to place the parameter sets included therein as well as the attributes therein as management objects into the above described management tree, an additional data element is determined to the management tree in accordance with the invention indicating the order of the different management objects and the priority among them.

[0023] Then, the management server S at first determines the contents of a new WAP provisioning document, and thereafter determines a data element for the parameter sets included in the WAP provisioning document and for the attributes included therein, the data element indicating the order of the different management objects and the priority among them, whereafter the management server updates the management tree using the data element concerned. After this, the management server sends the WAP provisioning document to the terminal TE. In response to the message of the management server S, the terminal TE deassembles the received WAP provisioning document, then updates the management tree in accordance with the data element indicating the order of the management objects and the priority among them, and stores the contents of the WAP provisioning document so that the definitions of the data element are taken into account. The terminal TE then includes the required WAP configuration, which allows using the WAP services without the user having to set the settings manually. The configuration can later be changed by pointing at it with a defined tag. For a more detailed description of the data elements included in the WAP provisioning document, reference is made to WAP Forum WAP specification WAP-292-ProvCont-20020815-d *"Provisioning Content, Draft Version, 12 October 2002"*.

[0024] Attaching the above described data element to the management tree can preferably be carried out in at least two different ways. Figure 4 illustrates the method according to a first preferred embodiment for connecting the data element to the management tree. When the WAP provisioning settings are determined in the management server, contents (400) of a new WAP provisioning document is at first determined. The WAP provisioning document may comprise several object types, such as interior objects, leaf objects or link objects. The sub-elements included in the WAP provisioning document, referred to as attributes, can be added to the management tree as specific nodes and/or leaves. Information is thus determined for the management objects or for some of the management objects of the WAP provisioning document thus determined, said information indicating the order of the different management objects and the priority (402) among them.

[0025] In this embodiment, information is determined as separate leaf objects (404) placed in the management tree in parallel at the same level in the management tree as the management object, whose priority they determine (cf. Figure 1). Such Rank leaf objects indicate the priority of a parallel management object to the other management objects, which parallel management object may therefore be for instance an individual parameter, a subtree or a parameter set. It should be noted that a priority does not necessarily have to be determined for all management objects in a WAP provisioning document; instead, the contents of the WAP provisioning document are such that the priority data should be determined for only some or for only one of the management objects.

[0026] After this, the management tree that is based on the WAP provisioning document can be updated (or if it does not exist, it can be formed) so that it comprises the Rank leaf objects (406) required for determining the priority of the management objects. The management server S thereafter sends the WAP provisioning document comprising the Rank leaf objects to the terminal TE (408). Here, the mechanisms of the device management protocol can be utilized as well as the messages determined for it. As to a more detailed description of the SyncML device management protocol, reference is made to the specification of the SyncML organization "*SyncML Device Management Protocol*", version 1.1.1, 2 October 2002.

[0027] The terminal TE receives the WAP provisioning document and deassembles the information (410) therein. The terminal interprets the

Rank leaf objects included in the document and updates (or forms) thereafter the management tree in response to the order of the management objects determined by the Rank leaf objects and to a priority (412) among them. As the contents of the WAP provisioning document is stored in accordance (414) with the management tree, the terminal TE then comprises the required WAP configuration for using the WAP services.

[0028] Figure 5 illustrates a method according to a second preferred embodiment for attaching a data element to the management tree. What is determined at first is contents (500) of the new WAP provisioning document and the order of the different management objects of the WAP provisioning document and information (502) indicating the priority among them. These stages correspond with stages 400, 402 in Figure 4.

[0029] In this embodiment, said information is determined as a portion of meta data (504) in the management tree, whereby the priority data can be modelled as a run-time property. In SyncML device management the meta data of the management tree can be provided with an optional run-time properties (RTProperties) definition comprising different run-time definitions of the management objects. As regards the details of the RTProperties elements, reference is made to the specification of the SyncML organization "*SyncML Device Management Tree and Description*", version 1.1.1, 2 October 2002, in which a new field "Priorities" can be added in accordance with the embodiment into the RTProperties fields described in chapter 7, and the contents of said field can be determined as illustrated above.

[0030] Then, the management tree that is based on the WAP provisioning document can be updated (or if it does not exist, it can be formed) so that it comprises RTProperties definitions (506) attached to the meta data of the management tree required for the determination of the priority of the management objects. The management server S then sends the WAP provisioning document to the customer device TE (508).

[0031] The terminal TE receives the WAP provisioning document and deassembles the information (510) therein. The terminal interprets the RTProperties definitions included in the meta data of the management tree and updates (or forms) thereafter the management tree in response to the order of the management objects determined by the RTProperties definitions and to a priority (512) among them. Finally, the contents of the WAP provisioning

document are stored in accordance (514) with the management tree to the terminal TE.

[0032] According to a preferred embodiment, the priority order of the attributes in the WAP provisioning document can be determined so that alternative attributes are for instance named differently and the priority order of the names is predetermined.

[0033] The above-described embodiments show that as regards the management of the device, the terminal TE operates as a customer device and the server S as a management server. However, it is possible that the terminal TE carries out the functionalities of the management device, whereby another terminal or server S may function as the customer device. A typical situation is such that the terminal user has determined or modified himself/herself the WAP provisioning settings of the terminal. Thus, in order to ensure the operation of the management system, the changed WAP provisioning settings have to be updated for the server S, in which case the above-described embodiments are carried out in reverse (from the terminal to the server).

[0034] It is apparent for those skilled in the art that as technology advances the basic idea of the invention can be implemented in various ways. The invention and the embodiments thereof are thus not restricted to the above examples but may vary within the scope of the claims.